Politecnico di Milano
School of Industrial and Information Engineering

Foundations of Artificial Intelligence
January 18, 2023

Prof. Francesco Amigoni & Prof. Pierluca Lanzi

FAMILY NAME

FIRSTNAME

CODICE PERSONA/ID

GENERAL INSTRUCTIONS

- **This is a closed-book/closed-notes exam.**
- **Pencils are not allowed.**
- **Answers must be written inside the answer boxes designated for each problem.**
- **Answers must be legible and adequately motivated.**
- **The exam must be returned with all its original sheets.**
- **No sheet can be added.**
- **None of the sheets can be removed.**
- **Only non-programmable calculators are allowed.**
- **Notes/books/mobile phones are not allowed.**
- **If a student is caught using forbidden material, the exam will immediately end, and the disciplinary committee will be notified.**
- **Exam receiving 5 points or less will be graded as REPEAT.**

SCORING

- **A problem left unsolved will amount to zero points.**
- **<u>Completely wrong</u> answers assign negative points.**

STUDENTS HAVE 1:30h TO SOLVE ALL THE PROBLEMS

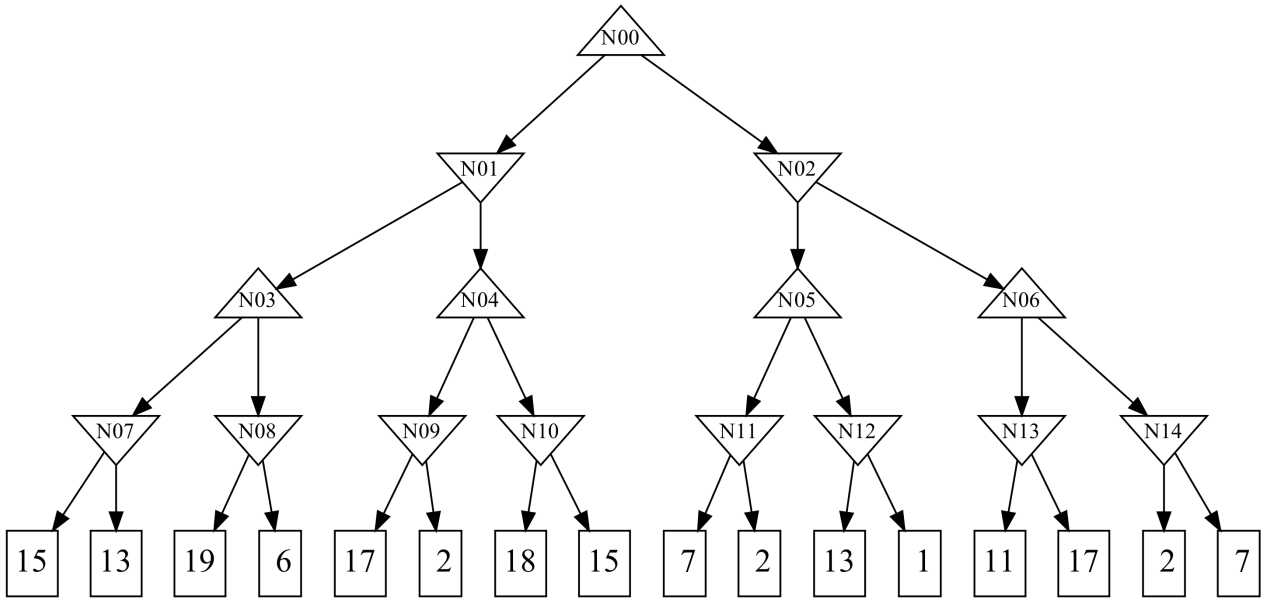
Problem Scores

/8	/8	/8
/8		

SIGN THIS BOX TO WITHDRAW FROM THE EXAM

00000

Adversarial Search. Consider the game tree depicted below.



Question 1: Apply the minimax algorithm and report the minimax value of each node. **Assume that children are generated and explored left to right.**

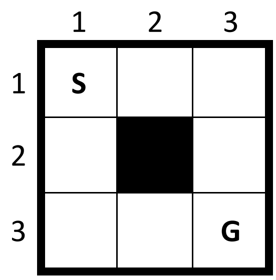
Node	Value		Node	Value
N00			**N08**	
N01			**N09**	
N02			**N10**	
N03			**N11**	
N04			**N12**	
N05			**N13**	
N06			**N14**	
N07				

Question 2: Now, apply the alpha-beta pruning algorithm and, for each node, specify whether the node has been pruned. **Assume that children are generated and explored from left to right.**

Node	Was it pruned?		Node	Was it pruned?
N00			N08	
N01			N09	
N02			N10	
N03			N11	
N04			N12	
N05			N13	
N06			N14	
N07				

Leaves listed left to right	15	13	19	6	17	2	18	15	7	2	13	1	11	17	2	7
Was it pruned?																

Reinforcement Learning. Consider the deterministic grid environment depicted below in which black cells represent obstacles, **S** represents the start cell, and **G** represents the goal cell.



Cells are identified using a pair (r,c) representing the cell at row r and column c. For example, (2,1) identifies the cell below the start cell. An agent must learn how to reach the goal cell using Q-learning. The agent can move in four directions: North (up), East (right), South (down), West (left); the agent can occupy any empty cell, the start cell, and the goal cell; the agent is not allowed to move toward an obstacle or outside the environment: for example, when at cell (2,1) the agent cannot move West (left) nor East (right) but only North (up) toward the start cell or South (down) toward cell (3,1). The agent is trained using the Q-learning algorithm with a discount factor of 0.5 and a learning rate of 0.25; the reward function returns 0 when the agent reaches the goal cell and -1 otherwise; the Q-table is initialized with zero values.

Question 1: In the first training episode the agent will start in **S** and perform the actions: East (right), East (right), South (down), South (down); thus, reaching the goal. What are the values in the Q-table after this first episode?

	North	**East**	**South**	**West**
(1, 1)				
(1, 2)				
(1, 3)				
(2, 1)				
(2, 3)				
(3, 1)				
(3, 2)				

Question 2. Suppose that the company you work for does not like reinforcement learning and asks you to model the same navigation task as a search problem. Define the functions you need to solve the same navigation task using A*.

Question 3. Now suppose that the company asks you to solve the same problem using a local search like hill climbing. How would you model the same navigation task as a hill-climbing problem?

Logic (8 points). Consider the following Knowledge Base (KB):

1. $A \rightarrow B$
2. $C \rightarrow A$
3. $G \wedge E \rightarrow C$
4. $F \rightarrow D$
5. $D \wedge E \rightarrow C$
6. $H \rightarrow E$
7. $H \rightarrow F$
8. H

Question 1: Apply the Backward Chaining (BC) algorithm to derive B from the KB and report the tree generated by the algorithm in the *worst case*.

Question 2: According to your answer to Question 1, does $\mathrm{KB} \vdash_{\mathrm{BC}} \{B\}$ hold or not? And $\mathrm{KB} \vDash \{B\}$? Why?

Question 3: If you apply the resolution algorithm using the unit resolution strategy can you establish if $\mathrm{KB} \vDash \{B\}$ holds or not? Why?
(You are not requested to actually apply the resolution algorithm.)

Planning (8 points). Consider a setting with a robotic arm, two boxes, two books, and two tables. Initially, the hand of the robotic arm is empty and the boxes and the books are on the same table. The goal is to have at least one box on the other table. The robotic arm can pick up and put down boxes and books and move its hand from one table to another.

Question 1: Model the above setting in STRIPS (you can use the operator $\neq$) reporting the constants and the predicates you use, the initial state, the goal, and the action schemas.
Write any assumption you make for aspects that are not specified.

Question 2: Considering the formulation you provided in Question 1 and forward planning, how many actions are applicable to the initial state? Report their signatures.

Question 3: If you are asked to model the above setting in PDDL, do you expect to use fewer or more constants, predicates, and action schemas? Why?
(A PDDL model is not requested.)